

Supervised multiway factorization

Eric F. Lock

Division of Biostatistics, School of Public Health, University of Minnesota

Gen Li

Department of Biostatistics, Mailman School of Public Health,
Columbia University

September 13, 2016

Abstract

We describe a probabilistic PARAFAC/CANDECOMP (CP) factorization for multiway (i.e., tensor) data that incorporates auxiliary covariates, *SupCP*. SupCP generalizes the supervised singular value decomposition (SupSVD) for vector-valued observations, to allow for observations that have the form of a matrix or higher-order array. Such data are increasingly encountered in biomedical research and other fields. We describe a likelihood-based latent variable representation of the CP factorization, in which the latent variables are informed by additional covariates. We give conditions for identifiability, and develop an EM algorithm for simultaneous estimation of all model parameters. SupCP can be used for dimension reduction, capturing latent structures that are more accurate and interpretable due to covariate supervision. Moreover, SupCP specifies a full probability distribution for a multiway data observation with given covariate values, which can be used for predictive modeling. We conduct comprehensive simulations to evaluate the SupCP algorithm, and we apply it to a facial image database with facial descriptors (e.g., smiling / not smiling) as covariates. Software is available at <https://github.com/lockEF/SupCP>.

Keywords: Faces in the wild; Singular Value Decomposition; Latent variables; Parafac/Candecomp; Tensors; Dimension Reduction

1 Introduction

Data are often best represented as a *multiway* array, also called a *tensor*, which extends the familiar two-way data matrix (e.g., Samples \times Variables). multiway data are increasingly encountered in many application areas. For example, in biomedical applications molecular profiling data may be measured over multiple body regions, tissue types, or time points for the same sample set (e.g., Samples \times Molecules \times Time \times Tissue). A multiway representation is also useful for imaging modalities. In this article we describe an application to aligned color images of the faces of different individuals, from the Labeled Faces in the Wild database (Learned-Miller et al. 2016). In particular we consider images for 4000 individuals, each with pixel dimensions 90×90 , where each pixel contains three color intensities (red, green, blue); thus the data is given by a 4-way array of dimensions $4000 \times 90 \times 90 \times 3$ (Individual $\times X \times Y \times$ Color).

Unsupervised factorization methods for multiway data, which extend well-known methods such as the *singular value decomposition* (*SVD*) and *principal component analysis* (*PCA*) for a data matrix, are commonly used in many fields. See Kolda & Bader (2009) for an accessible and detailed survey of multiway factorization methods and their uses. A classical and straightforward method is the *PARAFAC/CANDECOMP* (*CP*) (Harshman 1970) decomposition, in which the data are approximated as a linear combination of rank-1 tensors. Alternative approaches include the *Tucker* decomposition (Tucker 1966), and the *population value decomposition* (*PVD*) for 3-way data (Crainiceanu et al. 2011). All of these approaches reconstruct the data using a small number of underlying patterns in each modality, and they are therefore useful for dimension reduction, de-noising, and exploratory analysis.

In many scenarios, additional variables of interest are available for multiway data objects. For example, Kumar et al. (2009) provide several attributes for the images in the Faces in the Wild database, which describe the individual (e.g., gender and race) or their expression (e.g., smiling / not smiling). There is a growing literature on analyzing the relationship between multiway data and additional variables via predictive models in which the multiway data are predictors and the additional variables are treated as outcomes. Zhou et al. (2013) propose a tensor regression model for a clinical outcome, in which the coeffi-

cients for multiway data are assumed to have the structure of a low-rank CP decomposition. Miranda et al. (2015) describe a Bayesian formulation for regression models with multiple outcome variables and multiway predictors, which is applied to a neuroimaging study. Multiple methods have been developed for the classification of multiway data (Bauckhage 2007, Tao et al. 2007, Lyu et al. 2016), extending well-known linear classification techniques under the assumption that model coefficients have a factorized structure.

In this article we introduce a probabilistic CP factorization for multiway data that incorporates additional variables, called *SupCP*. Under this approach the additional variables are not outcomes to be predicted from multiway data; rather, they are treated as auxiliary covariates to inform a CP model for multiway data. Our framework is analogous to a class of supervised factorization methods for two-way data (matrices), in which covariates inform a PCA/SVD model. Our framework most directly extends the *supervised SVD* (*SupSVD*) approach (Li, Yang, Nobel & Shen 2016), which has also been generalized to accommodate sparse or functional PCA models (Li, Yang, Nobel & Shen 2016) and non-parametric relations between the covariates and principal components (Fan et al. 2016).

As in matrix factorization techniques such as PCA, the goal of multiway factorization is to capture underlying patterns that explain variation in the data. It is often reasonable to assume that these patterns are at least partially driven by additional measured variables of interest (e.g., gender or expression for facial images). Therefore, supervising on these variables can improve both the accuracy and the interpretation of the results. This motivates the SupCP method, and to our knowledge there is no previous work wherein additional variables inform the factorization or dimension reduction of multiway data. Moreover, SupCP specifies a full probability distribution for a multiway data observation with given covariate values. It can therefore be used for modeling of a multiway outcome given vector-valued predictors (rather than predicting an outcome vector from multiway predictors), for which there is little existing methodology.

The rest of this manuscript is organized as follows. In Section 2 we introduce some algebraic notation and terminology for multiway factorization. In Section 3.1 we introduce the SupCP model, in Section 3.2 we give its full likelihood, and in Section 3.3 we describe its identifiability properties. In Section 4 we describe a modified EM-algorithm

for simultaneous estimation of all parameters in the model. In Section 5 we describe the connections between SupCP and other methods in more detail. In Section 6 we describe a comprehensive simulation study to evaluate the SupCP method, and in Section 7 we describe an application to facial image data from the Faces in the Wild database.

2 Notation and Preliminaries

Here we give some algebraic notation and terminology for multiway data that will suffice to describe the development in the following sections. For a general and more comprehensive introduction to multiway data and its factorization we recommend Kolda & Bader (2009).

Define a K -way array (i.e., a K th-order tensor) by $\mathbb{X} : d_1 \times \dots \times d_K$, where d_k is the dimension of the k th *mode*. The entries of the array are given by square brackets, $\mathbb{X}[i_1, \dots, i_K]$, where $i_k \in \{1, \dots, d_k\}$ for $k \in 1, \dots, K$. In this article we use script capital letters to denote arrays of order greater than 2, bold capital letters to denote matrices ($\mathbf{X} : d_1 \times d_2$), and bold lowercase letters to denote vectors ($\mathbf{v} : d \times 1$).

For vectors $\mathbf{v}_1, \dots, \mathbf{v}_K$ of length d_1, \dots, d_K , respectively, define the *outer product*

$$\mathbb{X} = \mathbf{v}_1 \circ \mathbf{v}_2 \cdots \circ \mathbf{v}_K$$

as the K -way array of dimensions $d_1 \times \dots \times d_K$, with entries

$$\mathbb{X}[i_1, \dots, i_K] = \prod_{k=1}^K \mathbf{v}_k[i_k].$$

An array given by the outer product of vectors has *rank* 1. For matrices $\mathbf{V}_1, \dots, \mathbf{V}_K$ of the same column dimension R , we introduce the notation

$$[\mathbf{V}_1, \dots, \mathbf{V}_K] = \sum_{r=1}^R \mathbf{v}_{1r} \circ \dots \circ \mathbf{v}_{Kr},$$

where \mathbf{v}_{kr} is the r th column of \mathbf{V}_k . This defines a rank R CP factorization. Often, the columns of each \mathbf{V}_k are scaled to have unit norm ($\|\mathbf{v}_{kr}\|_F = 1$ for $r = 1, \dots, R$, where $\|\cdot\|_F$ represents the Frobenius norm) and scalar weights λ_r are given for each rank-1 component:

$$\sum_{r=1}^R \lambda_r \cdot \mathbf{v}_{1r} \circ \dots \circ \mathbf{v}_{Kr}.$$

The rank R CP approximation to an array of higher-rank \mathbb{X} is typically estimated by minimizing the residual sum of squares

$$\|\mathbb{X} - \sum_{r=1}^R \lambda_r \cdot \mathbf{v}_{1r} \circ \cdots \circ \mathbf{v}_{Kr}\|_F^2. \quad (1)$$

From (1) it is easy to see how the CP factorization extends the SVD for arrays of order greater than 2. However, there are some important differences. In particular, for $K > 2$ it is not common to enforce orthogonality among the columns of \mathbf{V}_k because this restricts the solution space and is not necessary for identifiability. Also, while the dimensions of the row space and column space are equal for $K = 2$, there is no analogous symmetry for $K > 2$, and therefore the rank of \mathbb{X} may be greater than the dimension of each mode (i.e., d_1, \dots, d_K).

Finally, it is often useful to represent a multiway array in matrix form, wherein it is *unfolded* along a given mode. For this purpose we let $\mathbf{X}^{(k)} : d_k \times \left(\prod_{i \neq k} d_i\right)$ be the array \mathbb{X} arranged so that the j th column contains all values for j th *fiber* in the k th mode; specifically, for $k = 1$

$$\mathbf{X}^{(1)} \left[j, i_2 + \sum_{k=3}^K \left(\prod_{l=2}^{k-1} d_l \right) (i_k - 1) \right] = \mathbb{X}[j, i_2, \dots, i_K],$$

and we define $\mathbf{X}^{(k)}$ similarly for $k = 2, \dots, K$.

3 Model

3.1 Latent Variable Model

Suppose we observe a K -way data array (e.g., an image with modes $x \times y \times \text{color}$) for each of n observational units, hereafter called *samples*. The full data set can then be represented as a $(K + 1)$ -way array with samples as the first mode, $\mathbb{X} : n \times d_1 \times \cdots \times d_K$. In addition, we observe a q -dimensional vector of covariates for each sample (e.g., describable features for facial images), given by the matrix $\mathbf{Y} : n \times q$. Without loss of generality, we assume both \mathbb{X} and \mathbf{Y} are centered across samples.

The SupCP model extends the CP factorization for \mathbb{X} (1) to a generative likelihood model in which the covariates in \mathbf{Y} inform latent patterns for the sample mode. Specifically,

the rank R SupCP model is

$$\begin{aligned}\mathbb{X} &= \llbracket \mathbf{U}, \mathbf{V}_1, \dots, \mathbf{V}_k \rrbracket + \mathbb{E} \\ \mathbf{U} &= \mathbf{Y}\mathbf{B} + \mathbf{F},\end{aligned}\tag{2}$$

where

- $\mathbf{U} : n \times R$ is a latent score matrix for the samples,
- $\{\mathbf{V}_k : d_k \times R\}_{k=1}^K$ are loading matrices in different modes,
- $\mathbb{E} : n \times d_1 \times \dots \times d_k$ is an error array with independent Gaussian entries $N(0, \sigma_e^2)$,
- $\mathbf{B} : q \times R$ is a regression coefficient matrix for \mathbf{Y} on \mathbf{U} ,
- $\mathbf{F} : n \times R$ has independent and identically distributed (iid) rows with mean 0 and $R \times R$ covariance Σ_f .

The sample scores \mathbf{U} are decomposed into what is explained by the covariates ($\mathbf{Y}\mathbf{B}$) and not explained by the covariates (\mathbf{F}). Therefore, the SupCP model decomposes the variation in \mathbf{X} into variation explained by the covariates, structured variation not explained by the covariates, and independent residual noise; this is easily seen with the following reformulation of (2):

$$\mathbb{X} = \llbracket \mathbf{Y}\mathbf{B}, \mathbf{V}_1, \dots, \mathbf{V}_k \rrbracket + \llbracket \mathbf{U}, \mathbf{V}_1, \dots, \mathbf{V}_k \rrbracket + \mathbb{E}.$$

The level of supervision on \mathbf{Y} depends on the size of the covariance Σ_f , relative to σ_e^2 . In particular, if $\|\Sigma_f\|_F$ goes to 0, it implies that all structured variation in \mathbb{X} is dependent on the variables in \mathbf{Y} . In Section 4 we describe a maximum likelihood estimation approach for all parameters simultaneously, including Σ_f and σ_e^2 ; therefore, the appropriate level of covariate supervision is determined empirically.

In practice, we constrain the columns of each \mathbf{V}_k to have a unit norm. The scaling of the factorization components, given by the λ_r 's in (1), are therefore subsumed into \mathbf{U} . Remarkably, this is enough to make the model parameters identifiable up to trivial sign changes and ordering of components, as discussed in Section 3.3.

3.2 Likelihood

The full marginal distribution for \mathbb{X} under the SupCP model, specified in (2), is a multilinear normal distribution (Ohlson et al. 2013). Here we derive its equivalent likelihood by considering the more tractable matricized version of \mathbb{X} , $\mathbf{X}^{(1)}$. The i th column of $\mathbf{X}^{(1)}$ gives all values for the i th sample. We define d as the overall dimension for each sample,

$$d = \prod_{k=1}^K d_k,$$

so $\mathbf{X}^{(1)}$ is of dimension $d \times n$. Further, let $\mathbf{V}_{\text{mat}} : d \times R$ be a matricized version of $\mathbf{V}_1, \dots, \mathbf{V}_K$, where the r th column is the vectorization of the K -way array given by $\mathbf{v}_{1r} \circ \dots \circ \mathbf{v}_{Kr}$. This gives a matrix representation of the SupCP model,

$$\mathbf{X}^{(1)} = \mathbf{Y} \mathbf{B} \mathbf{V}_{\text{mat}}^T + \mathbf{U} \mathbf{V}_{\text{mat}}^T + \mathbf{E}^{(1)}. \quad (3)$$

It follows that the distribution of $\mathbf{X}^{(1)}$, marginalizing over \mathbf{U} , is multivariate normal with log-likelihood

$$L(\mathbf{X}^{(1)}) \propto -\frac{n}{2} \log \det(\Sigma_{\mathbf{X}}) - \frac{1}{2} \text{tr}((\mathbf{X}^{(1)} - \mu_{\mathbf{X}}^T) \Sigma_{\mathbf{X}}^{-1} (\mathbf{X}^{(1)} - \mu_{\mathbf{X}})^T) \quad (4)$$

where

$$\Sigma_{\mathbf{X}} = \mathbf{V}_{\text{mat}} \Sigma_f \mathbf{V}_{\text{mat}}^T + \sigma_e^2 \mathbf{I}_d,$$

and

$$\mu_{\mathbf{X}} = \mathbf{Y} \mathbf{B} \mathbf{V}_{\text{mat}}^T,$$

and \mathbf{I}_d is a $d \times d$ identity matrix. Because the likelihoods of \mathbb{X} and $\mathbf{X}^{(1)}$ are equivalent, it suffices to consider (4). In Section 4 we describe an *Expectation-Maximization* (EM) algorithm to maximize this likelihood over $\{\mathbf{V}_1, \dots, \mathbf{V}_K, \Sigma_f, \sigma_e^2, \mathbf{B}\}$.

3.3 Identifiability

Here we describe conditions under which the parameters in the SupCP model are identifiable. That is, for two sets of parameters

$$\Theta = \{\mathbf{V}_1, \dots, \mathbf{V}_K, \Sigma_f, \sigma_e^2, \mathbf{B}\} \text{ and } \tilde{\Theta} = \{\tilde{\mathbf{V}}_1, \dots, \tilde{\mathbf{V}}_K, \tilde{\Sigma}_f, \tilde{\sigma}_e^2, \tilde{\mathbf{B}}\},$$

we give conditions for which equivalency of the likelihoods, $L(\mathbb{X} \mid \Theta) = L(\mathbb{X} \mid \tilde{\Theta})$, implies equivalency of the parameters, $\Theta = \tilde{\Theta}$.

Clearly, the model is not identifiable under arbitrary scaling of \mathbf{B} and the \mathbf{V}_k 's. For example, if $\{\Sigma_f, \sigma_e^2, \mathbf{B}\} = \{\tilde{\Sigma}_f, \tilde{\sigma}_e^2, \tilde{\mathbf{B}}\}$ and $\tilde{\mathbf{V}}_k = a_k \mathbf{V}_k$ for $k = 1, \dots, K$ where $\prod_{k=1}^K a_k = 1$, then $L(\mathbb{X} \mid \Theta) = L(\mathbb{X} \mid \tilde{\Theta})$. We address this by scaling the columns in \mathbf{V}_k to have unit norm, and restricting the first nonzero entry (typically the first entry) of each column to be positive:

- (a) $\|\mathbf{v}_{kr}\|_F = 1$, and $\mathbf{V}_k[1, r] > 0$ for $k = 1, \dots, K$ and $r = 1, \dots, R$.

The model is also clearly not identifiable under permutation of the R rank-1 components, i.e., applying the same re-ordering to the columns in $\mathbf{V}_1, \dots, \mathbf{V}_K$, and \mathbf{B} . This is easily addressed by rank-ordering the components under any given criteria; by default we order the components according to the diagonal entries of Σ_f , which will be distinct for non-trivial cases:

- (b) $\Sigma_f[r_1, r_1] > \Sigma_f[r_2, r_2]$ for $r_1 > r_2$.

After scaling and ordering, the model is identifiable under general conditions. It is remarkable that no other restrictions are required. In particular, the columns of \mathbf{V}_k need not be orthogonal and Σ_f need not be diagonal for identifiability.

There is a vast body of literature on the uniqueness of components in a CP factorization up to scaling and ordering (see Section 3.2 of Kolda & Bader (2009) for a review), which can be used to derive conditions for identifiability of the model given (a) and (b) above. Here we use a result of Sidiropoulos & Bro (2000), which requires the notion of k -rank. The k -rank of a matrix \mathbf{A} , $\text{kr}(\mathbf{A})$, is defined as the maximum number k such that any k columns of \mathbf{A} are linearly independent. Note that in general $\text{kr}(\mathbf{A}) \leq \text{rank}(\mathbf{A})$ and if \mathbf{A} is of full column rank, then $\text{kr}(\mathbf{A}) = \text{rank}(\mathbf{A})$. Sufficient conditions for identifiability of the SupCP model are given in Theorem 1.

Theorem 1 *For the model defined in (2), if*

$$\text{kr}(\mathbf{YB}) + \sum_{k=1}^K \text{kr}(\mathbf{V}_k) \geq 2R + K, \quad (5)$$

and if \mathbf{Y} is of full column rank, then the parameters $\{\mathbf{V}_1, \dots, \mathbf{V}_K, \Sigma_f, \sigma_e^2, \mathbf{B}\}$ are identifiable under the restrictions (a) and (b).

The proof of Theorem 1 is given in Appendix A. Generally we expect the solution to be matrices of full rank satisfying $\text{kr}(\mathbf{YB}) = \min(n, q, R)$ and $\text{kr}(\mathbf{V}_k) = \min(d_k, R)$, and therefore the model will be identifiable if R is sufficiently small relative to the dimensions of \mathbb{X} . Nevertheless, the conditions of Theorem 1 are easily verifiable. Moreover, these conditions are sufficient but not necessary, and so failure to satisfy (5) does not imply that the model is not identifiable.

In practice, although not necessary for identifiability, we also constrain the covariance Σ_f to be diagonal:

$$\Sigma_f[i, j] = 0 \text{ for } i \neq j.$$

This simplifies the model complexity considerably, and tends to improve performance of the estimation algorithm. We impose this constraint by default, but allow the option of an unconstrained covariance in our implementation.

4 Model Estimation

Here we describe estimation of all parameters for the model specified in Section 3.1. In Section 4.1 we describe a modified EM-algorithm for optimizing the likelihood (4) for fixed rank R . In Section 4.2 we discuss selection of the rank.

4.1 EM algorithm

For our implementation of the modified EM algorithm, the random factors \mathbf{U} are considered missing data to be imputed by their expected value at each iteration. For this purpose it is useful to consider the joint log-likelihood of \mathbb{X} and \mathbf{U} :

$$L(\mathbb{X}, \mathbf{U}) = L(\mathbb{X} \mid \mathbf{U}) + L(\mathbf{U}), \quad (6)$$

where

$$L(\mathbb{X} \mid \mathbf{U}) = -nd \log \sigma_e^2 + \sigma_e^{-2} \|\mathbb{X} - \llbracket \mathbf{U}, \mathbf{V}_1, \dots, \mathbf{V}_k \rrbracket\|_F^2, \quad (7)$$

and

$$L(\mathbf{U}) = -n \log \det \Sigma_f - \frac{1}{2} \text{tr} \left((\mathbf{U} - \mathbf{YB}) \Sigma_f^{-1} (\mathbf{U} - \mathbf{YB})^T \right).$$

Moreover, the conditional log-likelihood (7) can be expressed in terms of the matrices $\mathbf{X}^{(1)}$ and \mathbf{V}_{mat} used in Section 3.2:

$$L(\mathbb{X} \mid \mathbf{U}) = -nd \log \sigma_e^2 + \sigma_e^{-2} \text{tr} \left((\mathbf{X}^{(1)} - \mathbf{U} \mathbf{V}_{\text{mat}}^T) (\mathbf{X}^{(1)} - \mathbf{U} \mathbf{V}_{\text{mat}}^T)^T \right),$$

and so the joint likelihood (7) gives the sum of two multivariate normal likelihoods. This facilitates a modified EM algorithm, which is described in detail in steps 4.1.1 through 4.1.3.

4.1.1 Initialization

One could initialize \mathbf{U} and $\mathbf{V}_1, \dots, \mathbf{V}_K$ via a least-squares CP factorization (1), wherein the columns of each \mathbf{V}_k are scaled to have unit norms and the corresponding weights are absorbed into \mathbf{U} . However, this is estimated via an alternating least squares algorithm that can be computationally intensive, and is not guaranteed to find the global minimizer.

Alternatively we consider a random initialization in which the entries of each \mathbf{V}_k are generated independently from a normal distribution with mean 0, and then the columns of each \mathbf{V}_k are scaled to have unit norm, for $k = 1, \dots, K$. The latent factor \mathbf{U} is initialized by

$$\mathbf{U} = \mathbf{X}^{(1)} \mathbf{V}_{\text{mat}}^T.$$

In practice we find that random initialization performs as well as initialization via a CP factorization (see Appendix C), and so in our implementation we initialize randomly by default, with the option to initialize via CP factorization.

After initializing \mathbf{U} and $\mathbf{V}_1, \dots, \mathbf{V}_K$, we initialize \mathbf{B} by regressing \mathbf{Y} on \mathbf{U} :

$$\mathbf{B} = (\mathbf{Y} \mathbf{Y}^T)^{-1} \mathbf{Y}^T \mathbf{U}.$$

We initialize σ_e^2 by the sample variance of the entries in the residual array $\mathbb{X} - \llbracket \mathbf{U}, \mathbf{V}_1, \dots, \mathbf{V}_K \rrbracket$, and initialize Σ_f by the diagonal entries of the covariance of the residuals for the random factors $\mathbf{F} = \mathbf{U} - \mathbf{YB}$:

$$\Sigma_f = \text{diag}(\mathbf{F}^T \mathbf{F} / n),$$

where $\text{diag}(\mathbf{A})$ sets the off-diagonal values of a square matrix \mathbf{A} to zero.

4.1.2 E-Step

We estimate \mathbf{U} via its conditional expectation, given the data $\{\mathbf{X}^{(1)}, \mathbf{Y}\}$ and fixed parameters:

$$\hat{\mathbf{U}} = (\sigma_e^2 \mathbf{Y} \mathbf{B} \Sigma_f + \mathbf{X}^{(1)T} \mathbf{V}_{\text{mat}}) (\mathbf{V}_{\text{mat}}^T \mathbf{V}_{\text{mat}} + \sigma_e^2 \Sigma_f^{-1})^{-1}.$$

Similarly, the conditional variance of \mathbf{U} is

$$\hat{\Sigma}_U = \left(\frac{1}{\sigma_e^2} \mathbf{V}_{\text{mat}}^T \mathbf{V}_{\text{mat}} + \Sigma_f^{-1} \right)^{-1}.$$

We note that the matrix inverse can be efficiently evaluated via the Woodbury matrix identity (Press et al. 1986). Thus, the computation of the E-step is very efficient.

4.1.3 M-Step

We update the CP loading matrices $\mathbf{V}_1, \dots, \mathbf{V}_K$ via optimizing the expectation of the joint log-likelihood (6), using the expected moments of \mathbf{U} from 4.1.2. For instance, to update \mathbf{V}_1 , we fix $\mathbf{V}_2, \dots, \mathbf{V}_K$ and set the partial derivative of the log-likelihood with respect to \mathbf{V}_1 equal to 0. It helps to define $\mathbf{W}_{\text{mat}}^{(1)} : \left(n \prod_{k=2}^K d_k \right) \times R$ as the design matrix for regressing $\mathbf{X}^{(1)}$ on \mathbf{V}_1 . Given $\hat{\mathbf{u}}_r$ is the r th column of $\hat{\mathbf{U}}$, the r th column of $\mathbf{W}_{\text{mat}}^{(1)}$ is the vectorization of $\hat{\mathbf{u}}_r \circ \mathbf{v}_{2r} \circ \dots \circ \mathbf{v}_{Kr}$. We also let $\mathbf{V}_{\text{mat}}^{(1)} : \prod_{k=2}^K d_k \times R$ be the matrix with column r being the vectorization of $\mathbf{v}_{2r} \circ \dots \circ \mathbf{v}_{Kr}$. Then, \mathbf{V}_1 is given by

$$\mathbf{V}_1 = \mathbf{X}^{(1)T} \mathbf{W}_{\text{mat}}^{(1)} ((\mathbf{V}_{\text{mat}}^{(1)T} \mathbf{V}_{\text{mat}}^{(1)}) \cdot (\hat{\mathbf{U}}^T \hat{\mathbf{U}} + n \hat{\Sigma}_U))^{-1}$$

where \cdot is the Hadamard (entry-wise) product. The remaining loading matrices $\mathbf{V}_2, \dots, \mathbf{V}_K$ are updated similarly.

We then update the parameters \mathbf{B} , Σ_f , and σ_e^2 by setting their respective partial derivative of the expected log-likelihood equal to 0. As a result, we obtain:

$$\begin{aligned} \mathbf{B} &= (\mathbf{Y}^T \mathbf{Y}) \mathbf{Y}^T \hat{\mathbf{U}}, \\ \Sigma_f &= \frac{1}{n} \left(\hat{\mathbf{U}}^T \hat{\mathbf{U}} + n \hat{\Sigma}_U + (\mathbf{Y} \mathbf{B})^T (\mathbf{Y} \mathbf{B}) - (\mathbf{Y} \mathbf{B})^T \hat{\mathbf{U}} - \hat{\mathbf{U}}^T (\mathbf{Y} \mathbf{B}) \right), \\ \sigma_e^2 &= \frac{1}{nd} (\text{tr}(\mathbf{X}^{(1)T} (\mathbf{X}^{(1)} - 2 \mathbf{V}_{\text{mat}} \hat{\mathbf{U}}^T)) + n (\mathbf{V}_{\text{mat}}^T \mathbf{V}_{\text{mat}} \hat{\Sigma}_U) + \text{tr}(\hat{\mathbf{U}} \mathbf{V}_{\text{mat}}^T \mathbf{V}_{\text{mat}} \hat{\mathbf{U}}^T)). \end{aligned}$$

As a final step, we adjust the sign and scale of each column of \mathbf{V}_k and re-scale \mathbf{B} and Σ_f accordingly. This step is only for identifiability purpose (see Section 3.3) and does

not change the likelihood. Specifically, if $\tilde{\mathbf{V}}_1, \dots, \tilde{\mathbf{V}}_K, \tilde{\mathbf{V}}_{\text{mat}}, \tilde{\mathbf{B}}$ and $\tilde{\Sigma}_f$ correspond to their respective unscaled parameters, the scaled versions are given by

$$\begin{aligned} \mathbf{v}_{kr} &= \tilde{\mathbf{v}}_{kr} / \|\tilde{\mathbf{v}}_{kr}\|_F && \text{for } k = 1, \dots, K \text{ and } r = 1, \dots, R, \\ \mathbf{B}[i, r] &= \tilde{\mathbf{B}}[i, r] \cdot \|\tilde{\mathbf{v}}_{\text{mat}_r}\|_F && \text{for } i = 1, \dots, q \text{ and } r = 1, \dots, R, \\ \Sigma_f[r, s] &= \tilde{\Sigma}_f[s, r] \cdot \|\tilde{\mathbf{v}}_{\text{mat}_s}\|_F \cdot \|\tilde{\mathbf{v}}_{\text{mat}_r}\|_F && \text{for } s = 1, \dots, R \text{ and } r = 1, \dots, R. \end{aligned}$$

Optionally, we also restrict Σ_f to be diagonal by setting $\Sigma_f[r, s] = 0$ for $r \neq s$.

We iterate between the E-Step and the M-step of the algorithm until the marginal likelihood (4) converges. The M-step (Section 4.1.3) maximizes the expected log-likelihood coordinate-wise over each parameter, but may not give a global optimum over the full parameter space $\{\mathbf{V}_1, \dots, \mathbf{V}_K, \Sigma_f, \sigma_e^2, \mathbf{B}\}$. Nevertheless, the marginal likelihood is guaranteed to increase over each iteration and therefore converge. In practice the likelihood may converge to a local, rather than a global, optimum, depending on the initial values. We find that a form of annealing, in which random variation is incorporated into the first several iterations, helps to avoid dependence on the initialization. See Appendix C for more detail.

4.2 Rank selection

We recommend selecting the rank R of the model by likelihood cross-validation, in which model parameters are estimated via a training set and evaluated using the likelihood of a test set. Such an approach is desired because it directly assesses the fit of the specified probabilistic model (via the likelihood) in a way that is robust to overfitting. Also for exploratory purposes it provides validation that the estimated components describe real underlying structure.

First, we select n_{train} samples to fit the model, yielding the reduced data array $\mathbb{X}_{\text{train}} : n_{\text{train}} \times d_1 \times \dots \times d_K$ and covariate matrix $\mathbf{Y}_{\text{train}} : n_{\text{train}} \times q$. The remaining samples are considered the test set, yielding $\mathbb{X}_{\text{test}} : (n - n_{\text{train}}) \times d_1 \times \dots \times d_K$ and $\mathbf{Y}_{\text{test}} : (n - n_{\text{train}}) \times q$. For candidate values of R , we optimize the likelihood of the training data $\mathbb{X}_{\text{train}}$ and $\mathbf{Y}_{\text{train}}$ using the algorithm in Section 4.1 to obtain estimates $\{\hat{\mathbf{V}}_{1,R}, \dots, \hat{\mathbf{V}}_{K,R}, \hat{\Sigma}_{f,R}, \hat{\sigma}_{e,R}^2, \hat{\mathbf{B}}_R\}$.

The resulting estimates are assessed via the log-likelihood of the test set

$$L\left(\mathbb{X}_{\text{test}}, \mathbf{Y}_{\text{test}} \mid \hat{\mathbf{V}}_{1,R}, \dots, \hat{\mathbf{V}}_{K,R}, \hat{\Sigma}_{f,R}, \hat{\sigma}_e^2, \hat{\mathbf{B}}_R\right)$$

which can be evaluated as in (4). The rank is chosen to be the value of R that gives the lowest log-likelihood of the test set. The above approach works well in simulation studies, as shown in Appendix B.

5 Relationship to Other Methods

The SupCP model specified in Section 3.1 directly extends the SupSVD model (Li, Yang, Nobel & Shen 2016) when $K = 1$, i.e., when \mathbb{X} is a matrix. Thus, an alternative strategy is to apply SupSVD to the matricized data $\mathbf{X}^{(1)}$. However, this strategy is inefficient and therefore less accurate when the data have multi-linear structure, as shown in Sections 6 and 7; moreover, the patterns within each mode (\mathbf{V}_k) that are provided by a multiway factorization framework are often useful to interpret. Lastly, the identifiability results in Section 3.3 are not applicable to matrices; the SupSVD model requires additional restrictions of orthogonality and diagonal covariance for identifiability.

Without the presence of covariates \mathbf{Y} , SupCP reduces to a probabilistic CP factorization in which factors for the first mode (samples) are considered random. The EM algorithm in Section 4 fits such a model when $\mathbf{Y} = \mathbf{0}$. To our knowledge this particular unsupervised model and estimation approach, which provides a generative model for the sampled mode, is novel. Related maximum-likelihood based CP factorizations have been proposed by Mayekawa (1987) and Vega-Montoto & Wentzell (2003), and there is a large body of work on Bayesian models for the CP and other multiway factorizations (see, e.g., Hoff et al. (2011) and Zhou et al. (2015)).

If the entries of the residual factor covariance Σ_f are large relative to the noise variance σ_e^2 , then $\mathbf{U}, \mathbf{V}_1, \dots, \mathbf{V}_K$ reduce to a standard unsupervised least-squares CP factorization. Specifically, if Σ_f is diagonal and

$$\min_r \Sigma_f[r, r] / \sigma_e^2 \rightarrow 0,$$

it is easy to see that the joint likelihood (6) is maximized by the solution that minimizes

the squared residuals for the entries of \mathbb{X} ; the supervision component YB is given by a regression of Y and U , but will not influence the factorization of \mathbb{X} .

If the entries of the residual factor covariance Σ_f are small,

$$\|\Sigma_f\|_F^2/\sigma_e^2 \rightarrow 0,$$

then the residuals \mathbf{F} also tend to zero and the underlying structure of \mathbb{X} is given entirely by \mathbf{Y} :

$$\mathbb{X} = \llbracket \mathbf{YB}, \mathbf{V}_1, \dots, \mathbf{V}_k \rrbracket + \mathbb{E}. \quad (8)$$

This can be considered a multi-linear regression model for a multiway response (\mathbb{X}) from vector valued predictors \mathbf{Y} . This has received considerably less attention than the reverse scenario, prediction of a vector outcome from multiway predictors. However, recently Li & Zhang (2016) proposed a tensor response regression, wherein the tensor response is assumed to have a Tucker factorization with weights determined by vector-valued predictors. Hoff et al. (2015), describes a general framework for tensor-on-tensor regression, in which the predictor tensor and the outcome tensor have the same number of modes.

6 Simulation

Here we describe a comprehensive simulation study which compares three different methods: the proposed SupCP method, the least-squares CP factorization, and the SupSVD method (on matricized data along the first mode). We simulate data from 4 settings, and evaluate the performance of different methods in terms of parameter estimation and low-rank signal recovery accuracy. To avoid complication, we set the ranks for different methods to be the true ranks. The simulation study for rank estimation is conducted separately in Appendix B.

6.1 Simulation Settings

We consider a 3-way array \mathbb{X} with $n = 100$ samples, and d_1 and d_2 variables in the other two modes respectively. In particular, the generative model for \mathbb{X} is $\mathbb{X} = \llbracket \mathbf{U}, \mathbf{V}_1, \mathbf{V}_2 \rrbracket + \mathbb{E}$,

where $\llbracket \mathbf{U}, \mathbf{V}_1, \mathbf{V}_2 \rrbracket$ is the underlying low-rank signal with true rank $R = 5$, and \mathbb{E} is the Gaussian noise with iid entries with mean 0 and variance σ_e^2 . The auxiliary data matrix \mathbf{Y} contains $q = 10$ variables, potentially related to the latent variables in \mathbf{U} through a linear relation $\mathbf{U} = \mathbf{Y}\mathbf{B} + \mathbf{F}$. The coefficient \mathbf{B} is a 10×5 matrix, and the random matrix \mathbf{F} has iid rows with mean 0 and covariance Σ_f .

We particularly focus on the following 4 settings:

- **Setting 1 (CP Setting):** We set \mathbf{B} to be a zero matrix, indicating the auxiliary data have no effect on the underlying structure of \mathbb{X} (i.e., \mathbb{X} is generated from a probabilistic CP model). Both dimensions d_1 and d_2 are set to be 10, and the loading matrices \mathbf{V}_1 and \mathbf{V}_2 are normalized to have orthonormal columns respectively. The covariance matrix Σ_f is a diagonal matrix with diagonal values $\{100, 64, 36, 16, 4\}$ and $\sigma_e^2 = 1$.
- **Setting 2 (Collinearity Setting):** The coefficient matrix \mathbf{B} is filled with random numbers. The covariance matrix Σ_f is diagonal with diagonal values $\{25, 16, 9, 4, 1\}$. The signal-to-noise ratio, defined by $\|\mathbf{Y}\mathbf{B}\|_F / \|\mathbf{F}\|_F$, is around 1. The dimensions of the loading matrices are the same as those in Setting 1, but the unit-norm loading vectors are set to be highly collinear in \mathbf{V}_1 and \mathbf{V}_2 respectively. The noise variance is $\sigma_e^2 = 1$.
- **Setting 3 (Supervision Setting):** The coefficient matrix \mathbf{B} is the same as that in Setting 2, and the loading matrices are the same as those in Setting 1. The covariance matrix Σ_f is a zero matrix, and $\sigma_e^2 = 6$.
- **Setting 4 (High-dimension Setting):** Both d_1 and d_2 are set to be 50. The corresponding loading matrices are normalized to have orthonormal columns respectively. The coefficient matrix \mathbf{B} and the covariance matrix Σ_f are the same as those in Setting 2. The noise variance is $\sigma_e^2 = 0.5$.

We conduct 100 simulation runs for each setting, and compare different methods using various criteria. In particular, to evaluate the overall performance of dimension reduction, we assess the Frobenius norm of the difference between the estimated and true low-rank tensors $\|\llbracket \widehat{\mathbf{U}}, \widehat{\mathbf{V}}_1, \widehat{\mathbf{V}}_2 \rrbracket - \llbracket \mathbf{U}, \mathbf{V}_1, \mathbf{V}_2 \rrbracket\|_F$ (denoted by SE). We also consider the estimation

accuracy of loadings via the maximal principal angles $\angle(\mathbf{V}_1, \widehat{\mathbf{V}}_1)$ and $\angle(\mathbf{V}_2, \widehat{\mathbf{V}}_2)$ (Björck & Golub 1973). This only applies to SupCP and CP because SupSVD decomposes $\mathbf{X}^{(1)}$ and there is no direct estimation of \mathbf{V}_1 and \mathbf{V}_2 . In addition, we evaluate the estimation accuracy of other important parameters in SupCP and SupSVD via $\|\mathbf{B} - \widehat{\mathbf{B}}\|_F$, $(\sigma_e^2 - \widehat{\sigma}_e^2) / \sigma_e^2$ (relative error, denoted by RE_e), and the mean relative errors of the diagonal values of Σ_f (denoted by RE_f). We also compare the fitting times of different methods on a standard desktop computer (8Gb Ram, 3.30GHz).

6.2 Results

The comprehensive simulation results are presented in Table 1. Across all 4 simulation settings, SupCP always provides the highest low-rank signal estimation accuracy (i.e., smallest SE). Thus, from the dimension reduction point of view, SupCP outperforms the competing methods and identifies the most accurate underlying structure. In terms of fitting time, we observe that CP is the fastest among the three. Nevertheless, fitting of SupCP is also very fast: even in the High-dimension Setting, a SupCP model can be efficiently fitted in about 2 seconds on a standard desktop. The fitting time of SupSVD is sensitive to the dimensions of the data. This is because the number of parameters in SupSVD is proportional to the product of all dimensions, which grow rapidly with increasing dimensions.

In the CP Setting, where the auxiliary data \mathbf{Y} are irrelevant to the primary data \mathbf{X} , SupCP has similar loading estimation losses to CP. Thus, the proposed method is not affected by the uninformative data and automatically adapts to the unsupervised method in this setting. In addition, SupCP performs similarly to SupSVD in terms of other parameter estimations. We note the slightly worse performance of SupCP on RE_f is possibly due to its higher susceptibility to local optimum. See Appendix C for recommendations to address the local optimality issue. In the Collinearity Setting, SupCP almost uniformly outperforms the competing methods. In the Supervision Setting, where the auxiliary information is very important, both supervision methods (SupCP and SupSVD) outperform CP in signal recovery accuracy. Moreover, SupCP significantly improves the loading estimation over CP. Finally, in the High-dimension Setting, both tensor-based methods (SupCP and CP) outperform the matrix-based method (SupSVD) in terms of signal recovery accuracy and

computing time, because the numbers of parameters in SupCP and CP are significantly lower than that in SupSVD. In addition, SupCP is uniformly better than SupSVD in other criteria. Overall, the proposed method is computationally efficient and applicable to a range of settings.

7 Application to Facial Images

The Labeled Faces in the Wild database (Learned-Miller et al. 2016) is a publicly available set of over 13000 images, where each image includes the face of an individual. The images are taken from the internet and the individuals represented are celebrities. Each image is labeled with the name of the individual, but the images are unposed and exhibit wide variation in lighting, image quality, angle, etc. (hence “in the wild”).

Kumar et al. (2009) developed an attribute classifier, which gives describable attributes for a given facial image. These attributes include characteristics that describe the individual (e.g., gender, race, age), that describe their expression (e.g., smiling, frowning, eyes open), and that describe their accessories (e.g., glasses, make-up, jewelry). These attribute were determined on the Faces in the Wild dataset, as well as other facial image databases. In total 72 attributes are measured for each image. Our goal is to use these characteristics to supervise dimension reduction of the images and to develop a probabilistic model for computationally generating faces with given describable characteristics.

PCA and other low-rank matrix factorization approaches are commonly used to analyze facial image data. For example, the terminology *eigenfaces* (Sirovich & Kirby 1987, Turk & Pentland 1991) has been used to describe a broad class of methods in which PCA of vectorized facial images precedes a given task, such as face recognition. PCA can only be applied to vectorized images, wherein $d_1 \times d_2$ dimensional images are converted to a $d_1 \cdot d_2$ dimensional vectors (or $d_1 \cdot d_2 \cdot 3$ dimensional vectors for color images). Thus, PCA does not exploit multiway structure. Although facial images are not obviously multi-linear, the CP factorization has been shown to be much more efficient as a dimension reduction tool for facial images than PCA, and marginally more efficient than other multiway factorization techniques (Lock et al. 2011).

For our application the images are first *frontalized*, as described in Hassner et al. (2015).

That is, the unconstrained images are rotated, scaled, and cropped so that all faces appear forward-facing and the image shows only the face. After this processing step, we expect the nose, mouth and other facial features to be in approximately the same location across the images. This step is important for linear factorization approaches such as PCA or SupCP, which assume the coordinates are consistent across the images. Some examples of the frontalized images are shown in Figure 1.

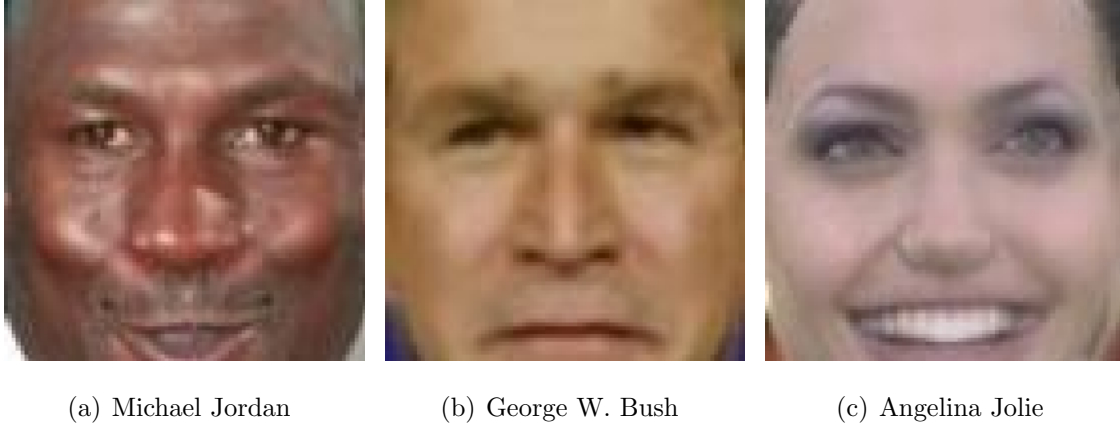


Figure 1: Examples of frontalized faces.

We work with a random sample of 4000 frontalized images from unique individuals. Each image is 90×90 pixels, and each pixel gives the intensity for colors red, green and blue, resulting in a multiway array of dimensions $\mathbb{X} : 4000 \times 90 \times 90 \times 3$. We center the array by subtracting the “mean face” from each image, i.e., we center each pixel triplet ($x \times y \times \text{color}$) to have mean 0 over the 4000 samples. The attribute matrix $\mathbf{Y} : 4000 \times 72$ is measured on a continuous scale; for example, for the smiling attribute, higher values correspond to a more obvious smile and lower values correspond to no smile. We standardize Y by subtracting the mean and dividing by the standard deviation for each row, hence converting each attribute to its z-scores.

We assess the fit of the SupCP model with a training set of 100 randomly sampled images. The mean log-likelihood for this training set and the mean log-likelihood for the remaining test images are shown for different ranks in Figure 2(a). The log-likelihood increases consistently for the training set, but the test log-likelihood peaks at a rank of 200, suggesting that over-fitting decreases the generalizability of the estimates for higher

ranks. For comparison, we also assess the SupSVD model on the vectorized images $\mathbf{X}^{(1)}$: 4000×24300 in Figure 2(b). The SupSVD model does not perform as well, achieving a maximum mean log-likelihood of -8.75×10^4 , versus a maximum of -7.55×10^4 for the SupCP model. Note that SupSVD requires many more parameters for each component, on the order of $90 \cdot 90 \cdot 3 = 24300$ parameters per component versus $90 + 90 + 3 = 183$ for SupCP. Thus, the SupSVD model is more prone to overfitting with a smaller number of components, which is inadequate to capture the complex covariance structure of the images and the covariate effects.

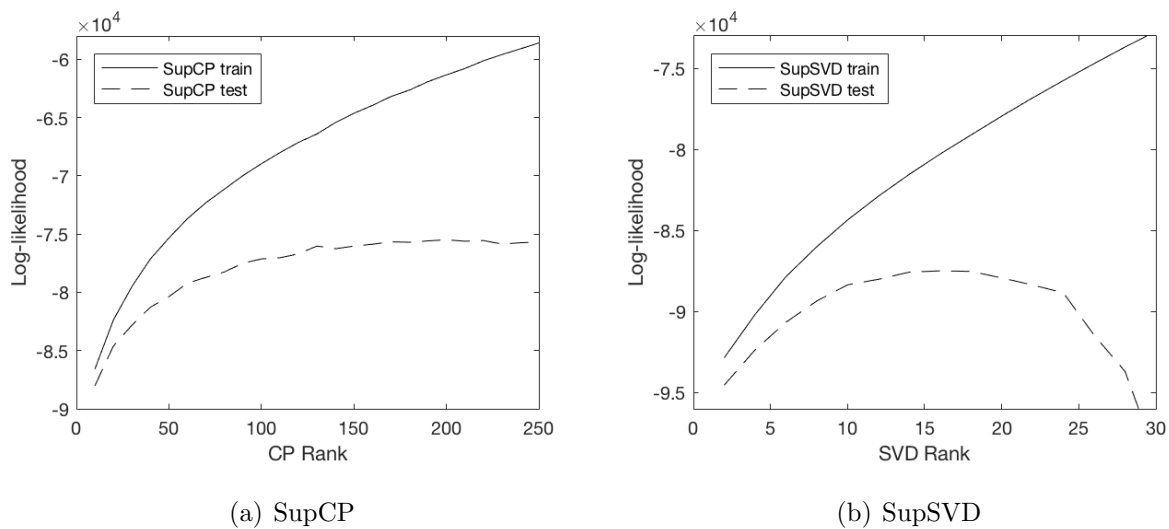


Figure 2: Log-likelihood for the estimated SupCP and SupSVD models, for different ranks, on the training images used to fit the model and the test images.

We fit the SupCP model, with rank 200, on the full set of 4000 images. This job required a computing server with 64GB of RAM. We run the EM algorithm for 2000 iterations, including a 500 annealing burn-in, which took approximately 8 hours. After 2000 iterations the log-likelihood appears to converge, as shown in Figure 3. The estimated parameters provide a data-driven generative model for faces with given covariates. In particular, given a feature vector \mathbf{y} with scores for the 72 describable attributes, the “mean face” for the given attributes is

$$[\mathbf{y}^T \mathbf{B}, \mathbf{V}_1, \mathbf{V}_2, \mathbf{V}_3],$$

where $\mathbf{V}_1 : 90 \times 200$ contains loadings in the x -dimension, $\mathbf{V}_2 : 90 \times 200$ contains loadings

in the y -dimension, and $\mathbf{V}_3 : 3 \times 200$ contains color loadings.

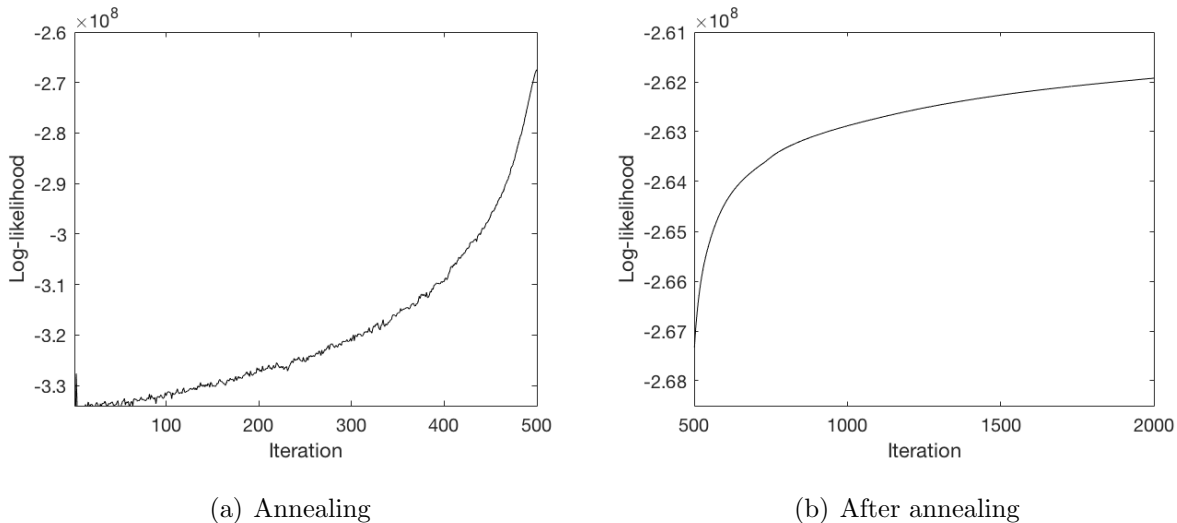


Figure 3: Log-likelihood for the model over 2000 EM iterations. The first 500 iterations (a) incorporate random annealing to avoid local modes.

The constructed mean facial images for certain attributes are given in Figure 4. To generate these images, the presence of a given features is coded as a z-score of 3 in the covariate vector, and all other scores are set to 0. The resulting images are smooth and remarkably intuitive.

8 Discussion

High-dimensional data with multiway structure are often encountered in modern research, yet present statistical methods to deal with such data are in their infancy and leave much room for further development. Most of the methodological development for the analysis of multiway data has come from fields other than statistics, such as computer science and chemometrics. Perhaps in part because of this, methods for multiway data are often not based on a generative statistical model. In this article we have developed the SupCP method, which is based on a probabilistic CP factorization model in which additional covariates inform the factorization. Our simulation studies illustrate the advantages of SupCP for dimension reduction and interpretation when the underlying structure of multiway data is multi-linear and at least partially driven by additional covariates. Our application to



Figure 4: Constructed faces under different attribute covariates.

facial image data illustrates how SupCP to estimate a generative model for multiway data with given covariates.

The SupCP framework introduced in this article may be extended in several ways. Here we have focused on the CP factorization because it allows for a straightforward extension of the supervised SVD model and has been shown to be an efficient factorization for facial image data. Supervised and probabilistic approaches to the Tucker factorization, or other multiway dimension reduction techniques, are interesting directions for future work. Moreover, there is a growing body of work on multiway factorization methods for sparse or functional data (Allen 2012, 2013). The SupCP framework may be extended to accommodate sparse and functional data, or non-linear covariate effects, which are analogous to recent extensions of the SupSVD framework (Li, Shen & Huang 2016, Fan et al. 2016). Finally, here we have focused on allowing a random latent variable and covariate supervision

for one mode (the sample mode). More general approaches which allow for supervision or random components in more than one mode is another promising direction of future work.

Acknowledgements

This work was supported in part by National Institutes of Health grant ULI RR033183/KL2 RR0333182 (to EFL).

A Proof of Theorem 1

Here we show that the parameters $\{\mathbf{V}_1, \dots, \mathbf{V}_K, \Sigma_f, \sigma_e^2, \mathbf{B}\}$ are identifiable given the likelihood (4), for the conditions of Theorem 1. By the identifiability of the mean and covariance of a multivariate normal likelihood, it suffices to show that any given mean

$$\mu_{\mathbf{X}} = \mathbf{Y}\mathbf{B}\mathbf{V}_{\text{mat}}^T$$

and covariance

$$\Sigma_{\mathbf{X}} = \mathbf{V}_{\text{mat}}\Sigma_f\mathbf{V}_{\text{mat}}^T + \sigma_e^2\mathbf{I}_d$$

uniquely identify $\mathbf{V}_1, \dots, \mathbf{V}_K, \Sigma_f, \sigma_e^2$, and \mathbf{B} .

Note the entries of $\mu_{\mathbf{X}}$ can be re-arranged as the $(K+1)$ -way array

$$\tilde{\mu}_{\mathbf{X}} = \mathbf{Y}\mathbf{B} \circ \mathbf{V}_1 \circ \dots \circ \mathbf{V}_K$$

of dimensions $n \times d_1 \times \dots \times d_K$. By the result of Sidiropoulos & Bro (2000), the components $\{\mathbf{Y}, \mathbf{B}, \mathbf{V}_1, \dots, \mathbf{V}_K\}$ are uniquely defined up to scaling and ordering if the k -rank condition (5) is satisfied. Hence, by the scaling and ordering restrictions (a) and (b), the components are fully identified. Also, the condition that \mathbf{Y} is of full column rank assures that \mathbf{B} is uniquely identified by

$$\mathbf{B} = (\mathbf{Y}\mathbf{Y}^T)^{-1}\mathbf{Y}^T\tilde{\mu}_{\mathbf{X}}.$$

Further, σ_e^2 is uniquely identified as the value that gives

$$\text{rank}(\Sigma_{\mathbf{X}} - \sigma_e^2\mathbf{I}_d) = R.$$

Letting

$$\Sigma_R = \Sigma_{\mathbf{X}} - \sigma_e^2 \mathbf{I}_d = \mathbf{V}_{\text{mat}} \Sigma_f \mathbf{V}_{\text{mat}}^T,$$

we see that Σ_f is uniquely identified by

$$\Sigma_f = (\mathbf{V}_{\text{mat}}^T \mathbf{V}_{\text{mat}})^{-1} \mathbf{V}_{\text{mat}}^T \Sigma_R \mathbf{V}_{\text{mat}} \mathbf{V}_{\text{mat}}^T \mathbf{V}_{\text{mat}}^{-1}.$$

□

B Rank Selection Simulation

Here we give the results of a simulation to assess the performance of the likelihood cross-validation approach to rank selection described in Section 4.2. The data simulation scheme is similar to that in Section 6.1, but the rank R is varied for each simulation. Specifically, data are generated for the 3-way array $\mathbb{X} : n \times d_1 \times d_2$ with $n = 100$ samples, $d_1 = 25$ and $d_2 = 25$, and \mathbf{Y} is an auxiliary matrix with 10 variables, $\mathbf{Y} : 100 \times 10$. Given rank R structure, data are generated under model (2) as follows:

1. The entries of \mathbf{Y} and $\mathbf{B} : 10 \times R$ are generated independently from a standard normal distribution.
2. The diagonal entries of the covariance $\Sigma_f : R \times R$ are generated from a $\text{Uniform}(5, 25)$ distribution, and the factor residuals $\mathbf{F} : n \times R$ have iid rows with mean 0 and covariance Σ_f .
3. The entries of $\mathbf{V}_1 : 25 \times R$ and $\mathbf{V}_2 : 25 \times R$ are generated to have orthonormal columns via the SVD of a random matrix.
4. The error array $\mathbb{E} : 10 \times 25 \times 25$ is given by independent $N(0, 1)$ entries.

We independently generate 10 datasets as above for each of $R = 1, \dots, 10$. For each of the $10 \cdot 10 = 100$ simulated datasets, we estimate the rank via the likelihood cross-validation scheme described in Section 4.2. We randomly specify a training set and a test set, each of size 50, and consider the ranks $r = 1, 2, \dots, 9$, or 10, choosing the rank that gives the lowest log-likelihood in the test set.

Figure 5 shows a scatterplot of the estimated rank and the true rank. Overall the estimated rank is a good predictor for the true rank; the correct ranks were chosen in 83 of the simulations, overestimated slightly in 13, and underestimated slightly for 4. This demonstrates that a likelihood cross-validation approach is reasonable to determine the rank of the model.

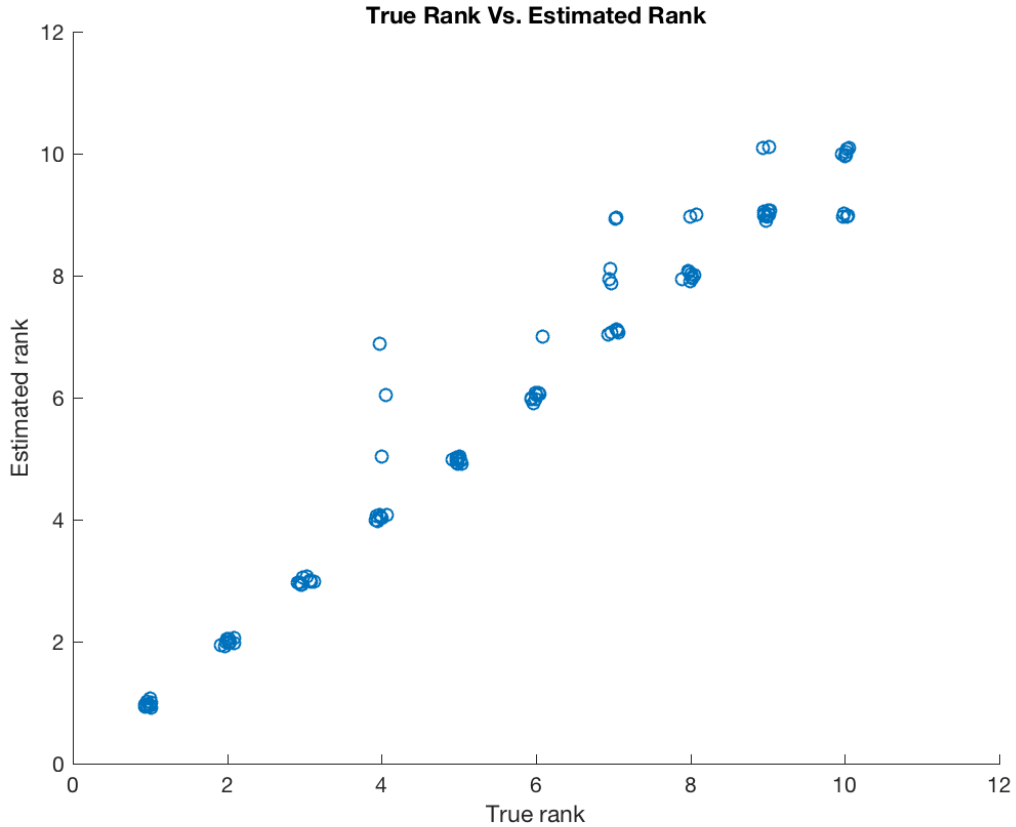


Figure 5: True rank vs. estimated rank for 100 randomly generated simulations. A small amount of jitter is added to show multiple points with the same coordinates.

C Initialization Simulation

Here we describe the results of a simulation to assess the sensitivity of the iterative algorithm in Section 4.1 to initial values. The algorithm can converge to a local maximum, and we explore different strategies to alleviate this issue.

Our simulation scheme is as follows. We consider a 4-way array \mathbb{X} of dimensions $10 \times 20 \times 40 \times 50$, and a vector of covariates $\mathbf{y} : 10 \times 1$. For rank $R = 2$,

1. The entries of $\mathbf{U} : 2 \times 10$ are generated independently from a normal distribution, variance of each row chosen from $\text{Uniform}(2, 22)$
2. The entries of \mathbf{y} are correlated with first column of \mathbf{U} ; specifically, $\mathbf{y} = \mathbf{u}_1 - \mathbf{f}$ where the entries of \mathbf{f} are generated from a standard normal distribution
3. The entries of $\mathbf{V}_1 : 20 \times 2$, $\mathbf{V}_2 : 40 \times 2$, $\mathbf{V}_3 : 50 \times 2$ are generated independently from a standard normal distribution and standardized so that the columns have unit norm.
4. The error array $\mathbb{E} : 10 \times 20 \times 40 \times 50$ is given by independent $N(0, 1)$ entries.

We generate 100 datasets under the above scheme, and for each dataset we run the algorithm under different initialization methods. We apply both random initialization and initialization via a CP decomposition, as described in Section 4.1.1. To avoid local maximum, we also consider a form of annealing in which random variation is incorporated during the first few iterations. Specifically, independent normally distributed noise with mean 0 is added to the expected value for \mathbf{U} at each iteration (Section 4.1.2), for the first L iterations, and the standard deviation of the noise is proportional to the inverse of the iteration number.

In total we consider six initialization methods: (1) random initialization with no annealing, (2) random initialization with 100 annealing iterations, (3) random initialization with 500 annealing iterations, (4) CP initialization with no annealing, (5) CP initialization with 100 annealing iterations, and (6) CP initialization with 500 annealing iterations. For each method, and each simulated data set, we run the algorithm until convergence twice under different random seeds.

In Table A1 we show the mean absolute difference in log-likelihoods between different runs, and the mean likelihood, under the different methods. Under this scheme, a CP initialization generally gives little benefit over a random initialization, and annealing improves the log-likelihood after convergence and the agreement under different runs. However, even 500 annealing iteration did not result in perfect agreement between different runs of the algorithm. To further reduce convergence to local optimum, one can run the algorithm

under multiple initialization and choose that which gives the highest log-likelihood after convergence.

References

- Allen, G. (2012), Sparse higher-order principal components analysis, *in* ‘International Conference on Artificial Intelligence and Statistics’, pp. 27–36.
- Allen, G. I. (2013), Multi-way functional principal components analysis, *in* ‘Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), 2013 IEEE 5th International Workshop on’, IEEE, pp. 220–223.
- Bauckhage, C. (2007), Robust tensor classifiers for color object recognition, *in* ‘Image Analysis and Recognition’, Springer, pp. 352–363.
- Björck, A. & Golub, G. H. (1973), ‘Numerical methods for computing angles between linear subspaces’, *Mathematics of computation* **27**(123), 579–594.
- Crainiceanu, C. M., Caffo, B. S., Luo, S., Zipunnikov, V. M. & Punjabi, N. M. (2011), ‘Population value decomposition, a framework for the analysis of image populations’, *Journal of the American Statistical Association* **106**(495), 775–790.
- Fan, J., Liao, Y., Wang, W. et al. (2016), ‘Projected principal component analysis in factor models’, *The Annals of Statistics* **44**(1), 219–254.
- Harshman, R. A. (1970), ‘Foundations of the PARAFAC procedure: Models and conditions for an “explanatory” multi-modal factor analysis’, *UCLA Working Papers in Phonetics* **16**, 1–84.
- Hassner, T., Harel, S., Paz, E. & Enbar, R. (2015), Effective face frontalization in unconstrained images, *in* ‘Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition’, pp. 4295–4304.
- Hoff, P. D. et al. (2011), ‘Separable covariance arrays via the Tucker product, with applications to multivariate relational data’, *Bayesian Analysis* **6**(2), 179–196.

- Hoff, P. D. et al. (2015), ‘Multilinear tensor regression for longitudinal relational data’, *The Annals of Applied Statistics* **9**(3), 1169–1193.
- Kolda, T. G. & Bader, B. W. (2009), ‘Tensor decompositions and applications’, *SIAM review* **51**(3), 455–500.
- Kumar, N., Berg, A. C., Belhumeur, P. N. & Nayar, S. K. (2009), Attribute and simile classifiers for face verification, in ‘2009 IEEE 12th International Conference on Computer Vision’, IEEE, pp. 365–372.
- Learned-Miller, E., Huang, G. B., RoyChowdhury, A., Li, H. & Hua, G. (2016), Labeled faces in the wild: A survey, in ‘Advances in Face Detection and Facial Image Analysis’, Springer, pp. 189–248.
- Li, G., Shen, H. & Huang, J. Z. (2016), ‘Supervised sparse and functional principal component analysis’, *Journal of Computational and Graphical Statistics* **25**(3), 859–878.
- Li, G., Yang, D., Nobel, A. B. & Shen, H. (2016), ‘Supervised singular value decomposition and its asymptotic properties’, *Journal of Multivariate Analysis* **146**, 7–17.
- Li, L. & Zhang, X. (2016), ‘Parsimonious tensor response regression’, *Journal of the American Statistical Association* (just-accepted).
- Lock, E. F., Nobel, A. B. & Marron, J. S. (2011), ‘Comment on Population Value Decomposition, a framework for the analysis of image populations’, *Journal of the American Statistical Association* **106**(495), 798–802.
- Lyu, T., Lock, E. F. & Eberly, L. E. (2016), ‘Discriminating sample groups with multi-way data’, *arXiv preprint arXiv:1606.08046*.
- Mayekawa, S. (1987), ‘Maximum likelihood solution to the PARAFAC model’, *Behaviormetrika* **14**(21), 45–63.
- Miranda, M., Zhu, H. & Ibrahim, J. G. (2015), ‘TPRM: Tensor partition regression models with applications in imaging biomarker detection’, *arXiv preprint arXiv:1505.05482*.

- Ohlson, M., Ahmad, M. R. & Von Rosen, D. (2013), ‘The multilinear normal distribution: Introduction and some basic properties’, *Journal of Multivariate Analysis* **113**, 37–47.
- Press, W. H., Flannery, B. P., Teukolsky, S. A. & Vetterling, W. T. (1986), ‘Numerical recipes: the art of scientific computing’, *Cambridge U. Press, Cambridge, MA* .
- Sidiropoulos, N. D. & Bro, R. (2000), ‘On the uniqueness of multilinear decomposition of N-way arrays’, *Journal of Chemometrics* **14**(3), 229–239.
- Sirovich, L. & Kirby, M. (1987), ‘Low-dimensional procedure for the characterization of human faces’, *Journal of the Optical Society of America A* **4**(3), 519–524.
- Tao, D., Li, X., Wu, X., Hu, W. & Maybank, S. J. (2007), ‘Supervised tensor learning’, *Knowledge and information systems* **13**(1), 1–42.
- Tucker, L. R. (1966), ‘Some mathematical notes on three-mode factor analysis’, *Psychometrika* **31**(3), 279–311.
- Turk, M. & Pentland, A. (1991), ‘Eigenfaces for recognition’, *Journal of cognitive neuroscience* **3**(1), 71–86.
- Vega-Montoto, L. & Wentzell, P. D. (2003), ‘Maximum likelihood parallel factor analysis (MLPARAFAC)’, *Journal of Chemometrics* **17**(4), 237–253.
- Zhou, H., Li, L. & Zhu, H. (2013), ‘Tensor regression with applications in neuroimaging data analysis’, *Journal of the American Statistical Association* **108**(502), 540–552.
- Zhou, J., Bhattacharya, A., Herring, A. H. & Dunson, D. B. (2015), ‘Bayesian factorizations of big sparse tensors’, *Journal of the American Statistical Association* **110**(512), 1562–1576.

Table 1: Simulation results under Setting 1, 2, 3 and 4 (each with 100 simulation runs). The median and median absolute deviation (MAD) of each criterion for each method are shown in the table. The best results are highlighted in bold.

Setting	Criterion	SupCP	CP	SupSVD
1: CP	SE	29.17 (1.33)	31.21 (1.04)	31.46 (0.60)
	$\angle(\mathbf{V}_1, \widehat{\mathbf{V}}_1)$	64.99 (22.21)	68.27 (16.16)	
	$\angle(\mathbf{V}_2, \widehat{\mathbf{V}}_2)$	52.14 (31.09)	63.66 (20.22)	
	$\ \mathbf{B} - \widehat{\mathbf{B}}\ _F$	47.19 (2.09)		46.35 (0.89)
	$100 * RE_e$	1.90 (1.03)		5.36 (0.99)
	$100 * RE_f$	19.28 (8.50)		10.23 (1.30)
	Time	0.50 (0.24)	0.03 (0.02)	0.06 (0.02)
2: Collinearity	SE	23.02 (0.78)	26.75 (0.95)	31.49 (0.60)
	$\angle(\mathbf{V}_1, \widehat{\mathbf{V}}_1)$	30.87 (12.88)	67.56 (12.97)	
	$\angle(\mathbf{V}_2, \widehat{\mathbf{V}}_2)$	71.63 (11.27)	72.27 (11.05)	
	$\ \mathbf{B} - \widehat{\mathbf{B}}\ _F$	40.80 (8.89)		65.55 (3.42)
	$100 * RE_e$	1.31 (0.76)		6.01 (1.05)
	$100 * RE_f$	28.84 (6.23)		24.46 (5.68)
	Time	0.62 (0.19)	0.03 (0.01)	0.25 (0.10)
3: Supervision	SE	34.43 (3.24)	74.54 (2.09)	68.03 (1.57)
	$\angle(\mathbf{V}_1, \widehat{\mathbf{V}}_1)$	25.25 (9.84)	69.67 (13.80)	
	$\angle(\mathbf{V}_2, \widehat{\mathbf{V}}_2)$	23.17 (7.47)	65.24 (12.84)	
	$\ \mathbf{B} - \widehat{\mathbf{B}}\ _F$	117.73 (12.47)		111.80 (9.82)
	$100 * RE_e$	1.77 (0.66)		8.36 (0.73)
	$100 * RE_f$	NA		NA
	Time	0.97 (0.13)	0.08 (0.03)	0.58 (0.25)
4: High-dimension	SE	22.34 (0.61)	23.46 (1.03)	83.31 (0.42)
	$\angle(\mathbf{V}_1, \widehat{\mathbf{V}}_1)$	13.00 (1.47)	13.75 (2.39)	
	$\angle(\mathbf{V}_2, \widehat{\mathbf{V}}_2)$	12.91 (1.26)	14.00 (2.66)	
	$\ \mathbf{B} - \widehat{\mathbf{B}}\ _F$	28.90 (0.84)		78.62 (11.06)
	$100 * RE_e$	0.20 (0.12)		5.13 (0.18)
	$100 * RE_f$	20.28 (2.61)		142.33 (16.73)
	Time	2.04 (0.68)	0.21 (0.11)	45.83 (12.94)

Table A1: Results under 6 different initialization approaches over 100 simulated datasets. The mean log-likelihood, mean absolute difference between runs for the same data, and mean computing time for a single run are shown, with standard deviations in parentheses.

Method	Log-likelihood	Difference	Time (sec)
Random start, no annealing	-200160 (589)	297.6 (576)	3.6 (3.1)
Random start, 100 annealing iters	-200090 (529)	180.8 (410)	4.7 (3.2)
Random start, 500 annealing iters	-199980 (420)	58.2 (132)	9.0 (2.6)
CP start, no annealing	-200320 (752)	433.5 (743)	3.1 (2.6)
CP start, 100 annealing iters	-200080 (501)	183.1 (404)	5.1 (3.3)
CP start, 500 annealing iters	-199980 (422)	53.6 (149)	9.6 (2.6)